```
*****************************************************************
*                                                               *
*                U N I V E R S I T Y   O F   U T A H            *
*                                                               *
*     C O M P U T E R   S C I E N C E   D E P A R T M E N T     *
*                                                               *
*****************************************************************

                    WEDNESDAY 06/11/80 AT 04:46:03 PM

                  L       EEEE PPPP  RRRR   EEEE  AAA  U   U
                  L      E     P   P R   R E     A   A U   U
                  L      E     P   P R   R E     A   A U   U
                  L      EEEE  PPPP  RRRR  EEEE  AAAAA U   U
                  L      E     P     R R   E     A   A U   U
                  L      E     P     R  R  E     A   A U   U
                   LLLL  EEEE P     R   R  EEEE A   A  UUU

TTTTT W   W  OOO   SSS   AAA  TTTTT        PPPP   AAA   SSS          1
  T   W   W O   O S   S A   A   T          P   P A   A S   S        11
  T   W   W O   O S     A   A   T          P   P A   A S             1
  T   W W W O   O  SSS  AAAAA   T          PPPP  AAAAA  SSS          1
  T   W W W O   O     S A   A   T          P     A   A     S         1
  T   W W W O   O S   S A   A   T    ..    P     A   A S   S  ..     1
  T    W W   OOO   SSS  A   A   T    ..    P     A   A SSS   ..   111
```

```
PROGRAM twosat;
(* check for satisfiability of a Boolean expression with two literals per clause *)
const
        max_variables = 50;
type
        senses = (uncomp, comp);
        clause = array[senses] of array[senses] of boolean;
        var_index = 1..max_variables;
var
        form:   array[1..max_variables] of array[1..max_variables] of clause;
        vars:   var_index;

procedure clearform;
(* initializes the form array to contain no clauses *)
var     i,j: var_index;
        s,t: senses;
begin
        for i := 1 to vars do
        for j := 1 to vars do
        for s := uncomp to como do
        for t := uncomp to comp do
                form[i][j][s][t] := false;
end;    (* clearform *)

procedure printliteral(i: var_index; s: senses);
begin
        if s = comp then write(tty, -i:2) else write(tty, i:2);
end;    (* printliteral *)

procedure printclause(i,j: var_index; s,t: senses);
begin   write(tty, '(');
        printliteral(i,s);
        write(tty, ' ');
        printliteral(j,t);
        write(tty, ')');
end;    (* printclause *)

procedure printform;
(* print the entire formula *)
var     i,j: var_index;
        s,t: senses;
begin
        writeln(tty);
        for i := 1 to vars do
                begin
                for j := i to vars do
                        for s := uncomp to comp do
                        for t := uncomp to comp do
                                if form[i][j][s][t]
                                        then    printclause(i,j,s,t);
                end;
        writeln(tty);
end;    (* printform *)

procedure insert(i,j: var_index; s,t: senses);
(* insert a clause in the formula *)
begin
        form[i][j][s][t] := true;
        form[j][i][t][s] := true;
end;    (* insert *)
```

```
procedure remove(i,j: var_index; s,t: senses);
(* remove a clause from the formula *)
begin
        form[i][j][s][t] := false;
        form[j][i][t][s] := false;
end;    (* remove *)

procedure informula;
(* read in the formula from the tty *)
var     i,j: integer;
        s,t: senses;
begin
        writeln(tty); break;
        writeln(tty, '*** TWO-SAT ***'); break;
        writeln(tty, 'enter number of variables: '); break;
        read(tty, vars);
        if (vars <= 0) or (vars > max_variables)
                then    begin
                        writeln(tty, 'error in number of variables (max ', max_variables:6); break;
                        end
                else    begin
                        writeln(tty, 'enter clauses, two integers per clause, - for negation, terminate with 0');
                        break;
                        clearform;
                        loop
                                read(tty,i);
                                exit if (i=0);
                                read(tty, j);
                                if i < 0
                                        then    if j < 0
                                                        then    insert(-i,-j,comp,comp)
                                                        else    insert(-i,j,comp,uncomp)
                                        else    if j < 0
                                                        then    insert(i,-j,uncomp,comp)
                                                        else    insert(i,j,uncomp,uncomp);
                                end;    (* loop *)
                        end;
end;    (* informula *)

function checksat: boolean;
(* check satisfiability of the formula *)
var     i,j: var_index;
        s,t: senses;
        in_form_comp, in_form_uncomp: boolean;

procedure cross;
(* forms the "cross product" of clauses with i complemented and uncomplemented *)
(* imports i,j, s, t *)
var     k: var_index;
begin
        if form[i][j][uncomp][s]
                then    for k := i to vars do
                                if form[i][k][comp][t]
                                        then    insert(j,k,s,t);
end;    (* cross *)

procedure delete;
(* delete clauses used in forming cross product *)
(* imports i,j, s, t *)
```

```
var     k: var_index;
begin
        if form[i][j][uncomp][s]
                then    begin
                        remove(i,j,uncomp,s);
                        for k := i to vars do
                                if form[i][k][comp][t]
                                        then    remove(i,k,comp,t);
                        end
end;    (* delete *)

begin   (* checksat *)
        checksat := true;       (* assume satisfiable until shown otherwise *)
        for i := 1 to vars do
                (* eliminate variable i *)
                begin
                (* does i occur complemented and uncomplemented by itself in clauses *)
                if form[i][i][comp][comp] and form[i][i][uncomp][uncomp]
                        then    checksat := false
                        else    begin
                                (* determine if literal i occurs in both complemented and uncomplemented form *)
                                in_form_uncomp := false;
                                in_form_comp := false;
                                for j := i to vars do
                                        begin
                                        if form[i][j][uncomp][uncomp] or form[i][j][uncomp][comp]
                                                then in_form_uncomp := true;
                                        if form[i][j][comp][uncomp] or form[i][j][comp][comp]
                                                then in_form_comp := true;
                                        end;

                                (* remove clauses or form cross product *)
                                if in_form_comp and in_form_uncomp
                                        then    begin
                                                (* add new clauses *)
                                                for j := i+1 to vars do
                                                        for s := uncomp to comp do
                                                        for t := uncomp to comp do
                                                                cross;
                                                end;
                                        (* delete old clauses *)
                                        for j := i to vars do
                                                for s := uncomp to comp do
                                                for t := uncomp to comp do
                                                        remove(i,j,s,t);
                                end;
                printform;
                end;
end;    (* checksat *)

begin   (* main program *)
while true do
        begin
        informula;      (* read in clause and put into form *)
        printform;
        writeln(tty); break;
        if checksat     (* check satisfiability *)
                then    begin write(tty, 'ans: satisfiable'); break; end
                else    begin write(tty, 'ans: unsatisfiable'); break; end;
        writeln(tty);
```

```
        end;
end.

[PHOTO:  Recording initiated  Thu 5-Jun-80 5:05PM]

LINK FROM KELLER, TTY 11

 TOPS-20 Command processor 4(557)
@passgo
LIST       =
SOURCE     = twosat.pas

PASSGO/LOTS: TWOSAT [ TWOSAT      ] PAGE  1..
   0 ERROR(S) DETECTED

TOTAL:   187 LINES

   COMPILE RUNTIME: 00:01.019      ELAPSED: 00:00:05.7

TWOSAT     : EXECUTION

TO CONTINUE, HIT THE RETURN KEY *

*** TWO-SAT ***
ENTER NUMBER OF VARIABLES:
5
ENTER CLAUSES, TWO INTEGERS PER CLAUSE, - FOR NEGATION, TERMINATE WITH 0
1 -2 1 4 ¬H$J¬H$J3 -1 4 -1 -5 5 -4 -3 4 0

( 1 -2)( 1  3)(-1  4)(-1 -5)(-3  4)(-4  5)


(-1 -5)(-2  4)(-2 -5)( 3  4)(-3  4)( 3 -5)(-4  5)

(-1 -5)( 3  4)(-3  4)( 3 -5)(-4  5)

(-1 -5)( 4  4)( 4 -5)(-4  5)

(-1 -5)( 5 -5)(-5  5)

(-1 -5)
ANS: SATISFIABLE

*** TWO-SAT ***
ENTER NUMBER OF VARIABLES:
¬C
@po

[PHOTO:  Recording terminated  Thu 5-Jun-80 5:06PM]
```

```
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
        USER: LEPREAU                   FILE NAME: TWOSAT.PAS.1         ACCOUNT: CS-CLASS               PAGES: 000006
**********************************************************************************************************************
```